Game Theoretic approach towards Optimal Multi-tasking and Data-distribution in IoT

Mo Haghighi, Konstantinos Maraslis, Theo Tryfonas, George Oikonomou, Alison Burrows, Pete Woznowski and Rob Piechocki
School of Engineering
University of Bristol

***Abstract*— Current applications of Internet of Things (IoT) often require nodes to implement logical decision-making on aggregated data, which involves more processing and wider interactions amongst network peers, resulting in higher energy consumption and shorter node lifetime. This paper presents a game theoretic approach used in Sensomax, an agent-based WSN middleware that facilitates seamless integration of mathematical functions in large-scale wireless sensor networks. In this context, we investigate game theoretic and auction-based techniques to optimise task distribution and energy consumption in IoT networks of multiple WSNs. We also demonstrate how our proposed game theoretic approach affects the performance of WSN applications with different operational paradigms.**

Keywords*— *Game theory; IoT; WSN; Energy-efficient; Sensomax;

I. Introduction

IoT (Internet of Things) has become a major technology enabler for a wide variety of applications ranging from medical, to military and environmental monitoring. These require data to be aggregated online, and often involve a number of actions to be applied on the environment (i.e. via actuators) as a result of the data aggregation process. They include multiple interconnected WSNs with a more sophisticated set of logical and mathematical functionalities, often implemented in a collaborative fashion amongst a large number of sensors and actuators. Sensor nodes therefore require a relatively powerful processor and sufficient memory for retrieving raw data and storing the processed ones, which in turn requires increasingly more resourceful components, but supplying sensors with larger energy sources often results in larger, harder to deploy sensors. Modern applications also attempt to maximise WSN's reusability by sharing their resources amongst multiple applications, in order to reduce the cost and the hardship involved in WSNs' deployment.

There have been several attempts to provide sensors with reliable software in order to regulate their energy consumption and manage applications' access to their underlying hardware. Middleware is a software layer, which lies between the application layer and the hardware components. It acts as an intermediary by dividing the available hardware functionalities based on their characteristics, and providing the applications with their required functionalities in the form of services. Provision of such services is implemented by maximising resources usability and meeting the application requirements, whilst minimising resources' energy footprint.

After introducing Sensomax, the next section will review existing research using game theory in WSNs. In section three, we describe the methodology by which game theory is used in Sensomax for the aforementioned purposes. Finally the case study section presents experiments conducted to validate our proposed game theoretic approach in optimising resource allocation and task distribution in WSN.

Sensomax [1-2] is an agent-based WSN middleware, which supports concurrent execution of multiple applications, integrates different mechanisms for different operational paradigms, and facilitates application developers with a component-based architecture for seamless development process. Sensomax is written in Java, and was modified to be used on various java-enabled hardware devices such as the Raspberry Pi. The modular architecture of Sensomax makes it easy to embed various logical and mathematical operations within the applications using Java APIs. It also provides developers with a set of pre-built APIs for modifications.

One of the outstanding features of Sensomax is a hierarchical communication mechanism, which abstracts the network into logical regions with exclusive functionalities. In Sensomax, each application is allocated a region according to its needs, whilst resources in the same region can be utilised by other applications simultaneously. However, each application's concurrent utilisation of the network resources is completely invisible to other applications. Such capability could also create a multi-tier and collaborative execution environment, where multiple applications' interaction using the same set of hardware resources is necessary.

In [18] we used game-theory in the same manner to facilitate energy-efficiency. In this paper however we will use auction-based techniques [3-4] with game theory to optimise multi-tasking. [3-4] used auction-based techniques to distribute applications' tasks amongst sensors based on their available resources, and investigated how quickly and energy-efficient applications requirements could be served.

II. Related Work

Game theory is a mathematical tool aimed at solving conflicts and encouraging cooperation among rational participants. Participants, who are known as 'players', tend to focus their decision-makings based on receiving the best possible reward [6]. Depending on the characteristics of a game, there are different approaches available to derive a solution:

- A game is ***Cooperative*** when maximising the overall payoff is the primary objective, and players are not concerned with individual payoffs; conversely, in ***Non-cooperative*** games, individual payoffs are more important than the overall one.
- A game is considered to be of ***Perfect Information***, when every player knows all the strategies, which have been followed by others earlier in the game; ***Imperfect Information*** indicates a lack of information about the other players moves earlier in the game.

- In ***Static*** (or one-shot) games, all players choose a strategy simultaneously at the beginning and they cannot change it throughout the game; ***Dynamic*** games allow players to change strategies while the game is in progress.
- A game in a ***Strategic form*** can be defined only by the strategy sets of the involved players, which are not known amongst players, and the function that determines the rewards given the followed strategies; an ***Extensive form*** game is sequential and often represented by a tree-graph, which provides players information about other players in addition to their sequence.
- ***Nash Equilibrium*** is a set that consists of the strategies of all players, called optimal strategies, and that leads to a payoff for each player such that none of them can unilaterally change their strategy and gain higher reward than before. If a set of mixed strategies leads to a Nash Equilibrium, then this is called a mixed Nash Equilibrium. A game can have more than one Pure Nash Equilibrium [4].
- A player's strategy is considered ***Dominant*** when it always leads to better rewards for a given player than any other available strategy.

Game theory can be used as a rational decision-making tool to solve conflicts of interest amongst network peers in WSNs. The kind of game played by network peers in Sensomax is a non-cooperative game with perfect information. According to [5], the main categories that the game theoretic approaches of such conflicts fall into are: *Network Management,* with indicative topics such as Resource Allocation, Task Scheduling and Power Control, *Communication* with topics like Quality of Service (QoS), Topology Optimization and Routing Protocol Design, *Network Security* grappling with Intrusion/Denial of Service Attack Detection and Prevention and finally *Applications* such as Target Tracking, Data Collection and Packet Forwarding. In [17] we conducted a thorough study of game theory for the security aspects of IoT.

In terms of network management, communication and applications, [15] offers a model to improve the performance of a heterogeneous WSN, by taking into consideration the reliability, connectivity and the power efficiency of the network. The results indicate that the existence of a Nash Equilibrium is always achievable. In similar work, [13] builds an energy-efficient control model, which offers great improvement to energy reduction in terms of QoS. it attempts to improve the so-called Gur Game algorithm [14], a mathematical model that is used for self-control in cooperative environments.

[10] offers a Localized Game theoretical Clustering Algorithm (LGCA), which tackles the problem of choosing the most appropriate cluster heads. It attempts to improve the Clustered ROuting for Selfish Sensors (CROSS) [11], and the Low-Energy Adaptive Clustering Hierarchy (LEACH) [12]. As the most fundamental part of the proposed solution, knowledge on the number of players (nodes) in each round is considered unnecessary. The key for this is that each node plays a clustering game only with its neighbours within a predefined radius. Moreover, exactly one node can bid for a position of the cluster-head in one district successfully, in order to achieve an optimal payoff. Simulation showed that LGCA performs better than CROSS and LEACH in terms of network lifetime. Focusing on security related applications of game theory within WSNs, [16] investigates cases where a clustered WSN is under attack. The proposed Intrusion Detection System (IDS) monitors the data transfers and strains to keep the WSN functioning properly. This situation is modelled as a two-player, non-cooperative, zero-sum game where the attacker's reward is proportional to the damage caused to the network and the defender, which is represented by the IDS, receives a reward proportional to the network's functionality. As a result, it is proved that the game has no Pareto optimal and no pure Nash Equilibrium.

[7] applies a reputation system on different sensors in order to make it more energy efficient and secure. Forwarding packages, in a fashion required by both ends, brings positive reputation to a sensor, but also consumes more energy, which could ultimately affect the networks performance later. There are malicious nodes that are injected in the network in order to randomly drop packets in order to shut down nodes with low reputation. In such cases there arises a number of conflicting motivations, where game theoretic tools could offer a suitable solution. This model, which extends the works done in [8-9], attempts to divide nodes' interaction into three distinctive domains including: any node-to-node communication; one-hop neighbours communication; and inner-cluster communication. It concludes that for all three types, there can always be a Nash equilibrium, by which security and power conservation can be improved.

In [10] there is a game theoretic approach of multiple collaborating intruders who try to inject malicious data into a target node and the "defender" (the IDS) tries to prevent the attack. Since intruders can be assumed to act as one, there is a two-player, non-cooperative zero sum game that occurs. Intruders, in their attempt to send their package try to find the paths leading to the target node that will maximize the probability of successful delivery. The IDS on the other hand, can opt among different sampling strategies aiming to minimize the probability of a successful attack always by taking into account the underlying cost of each sampling strategy. Under that scheme, the authors demonstrate the existence of a Nash Equilibrium and the optimal strategies.

III. METHODOLOGY

In the introduction section we explained why executing multiple applications is crucial to WSNs. We also briefly described how Sensomax is capable of multitasking, in order to serve multiple end-users simultaneously. In [18] we have demonstrated how game theory can facilitate energy-efficient task distribution using the same methodology. In this paper however, our focus will be on utilizing game theory for multi-tasking in WSNs. Multi-tasking capability is in fact implemented through hierarchical task distributions amongst multiple clusters. In this section we will describe how game theory is technically exploited within the Sensomax architecture for more optimal task distribution amongst clusters running multiple tasks.

Before we go deep into any low level details, it is worth mentioning that serving the end-users is considered the most important requirement of the base station. Therefore, the base station needs to satisfy the services required by the end-user as its first priority. However, new applications, which are deployed onto the base station, either by the existing end-users or the new ones, have certain requirement that also need to be satisfied in addition to the pre-deployed applications.

In the context of game theory, the base station should weight its strategies and choose the best option, which maximises or at the least maintains its profit (utility) whilst meeting old and new applications' requirements simultaneously. A node's utility is defined by the amount of processing time spent on the given tasks, where maximising utility means spending less processing, thus saving more energy.

As was briefly pointed out in the introduction section, the decision (strategy taken) of the base station will be known to the cluster-heads. The same applies to the communications between the cluster-heads and their members. Therefore in this section we will consider maintaining or maximising the base station, cluster-heads' and cluster members' utilities in an extended game theoretic form with perfect information. Since this case study only deals with the interactions between the base station and the cluster-heads, the decision maker needs to make its choice between the dominated strategy and the Nash equilibrium (if any) with prior knowledge on the decision taken by the peer one-level higher in hierarchy.

As we explained earlier, our proposed approach uses auction-based algorithm in conjunction with the game theory. The auction-based algorithms are only used to calculate the price of each task. More details, including the algorithms are given in [3] and [4]. When the base station receives an application from the end-user, it initially needs to query all the cluster-heads with the operational details of the new task, in order to collect their offers for the task. As was mentioned earlier, Cluster-heads advertise their offers based on their on-going operations and the number of applications running concurrently. Once all offers are collected, the base station starts the task distribution process by initially applying the game theoretic approaches in order to identify its options, or in other words, its strategies, as defined by the game theory.

Every application is comprised of a number of tasks, which can be priced based on its given operational paradigms. Therefore, each task has a certain value regardless of every node's operational state. Once applications arrive in the base station, their tasks are priced, and offers are collected from the cluster-heads.

For this case study, full details of the game will be given in order to establish a better understanding of how interaction works amongst the sensor nodes. Also the game is implemented in a simplified form with low network density.

All nodes are assumed to be indifferent in terms of their capabilities, their on-going operations, number of concurrent applications and their remaining energy level. Hence, for the first phase of this experiment we will demonstrate the interaction between the base station and a single cluster-head in order to identify their available strategies with perfect information. This interaction can be envisioned as a game between the base station and the cluster-head, where both network entities' rewards calculated based on the task price and the available resources in the node. In the second phase, the game will expand to include more nodes, thus creating a wider interaction amongst network entities.

The cluster-head used in this experiment is already running an application, which requires registering ***Temperature*** at 5-second intervals, and forwarding the recorded data to the base station at 60-second intervals. This application is hereafter referred to as the 'pre-deployed' application. Assuming that the base station receives a new application, which requires recording ***Light*** level, with the same timing and recording requirements as the previously deployed application, here we will analyse the interaction of the base station and the cluster-head in handling the new application.

The maximum utility (reward) of every cluster-head is achieved by minimising the processing time, thus saving more energy for longer lifetime, whereas the maximum utility of the base station is directly related to serving the end-users' application requirements.

The decision-makings done by the network entities (including the base station, cluster-heads and the nodes) for handling the new task, will narrow their choices down into prioritising their given tasks.

The base station can make its selection from the following strategies:

A. Receive the task- never relay it to any CH (Priority 0)
B. Accept the task- relay it immediately (Priority 1)
C. Accept the task - delay its relay with minor latency (Priority 2)
D. Accept the task and delay its relay with major latency (Priority 3)

The priority number appearing next to each option indicates the execution priorities based on Sensomax's internal architecture. Priorities define how urgent the tasks need to be executed, and effectively assign their position in the execution queue. Similarly, the cluster-head also has the above-mentioned options as the base station, except that, instead of relaying the given tasks to another node, it executes them internally, which results into the following strategies:

a. Receive the task and never execute it (Priority 0)
b. Accept and execute the task immediately (Priority 1)
c. Accept the task and delay its execution with minor latency (Priority 2)
d. Accept the task and delay its execution with major latency (Priority 3)

Given the above strategies for both the base station and the cluster-head, and assuming that the entities cannot reverse their decisions (static game), and the new application has a lower priority to the pre-deployed one, the interaction can be demonstrated as shown in figure 1. Rewards are shown in parentheses with the first figure denoting the base station's '**BS**' reward and the second standing for the cluster-head's '**CH**' reward, in the form of:

("Base station's reward", "Cluster-head's reward")

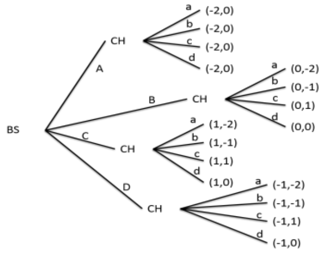


Figure 1: Base station and Cluster-head in a game-tree

As this figure shows, the base station's maximum reward leans towards taking strategy 'C' with payoff 1, whereas the Cluster-head's maximum rewards can be achieved by taking strategy 'c'. That is because, if the base station takes strategy

‘C’, it in fact serves the end-user’s requirements to the best of its capability and maintains serving the pre-deployed application as well, compared to relaying the task with minor or major delays, or not relaying the task at all. The same applies to the cluster-head, whereby taking strategy ‘c’, which executes the application with minor delay, allows it to first execute the pre-deployed task and then act on the new one. Whereas taking other strategies either delay the current task (b), never executes the new task (a), or executes the new task with a major delay (d). In case ‘a’, although not executing the task results in less energy consumption, the cluster-head is still acting against the base station requirements, and will be queried frequently for the given task’s progress (which results in spending energy on processing the queries), and the it could be assumed faulty by the base station and ultimately excluded from the network.

Given the above explanation, there is a single dominant strategy, which is also the Nash equilibrium: (C, c) resulting in maximum payoff for both network entities. This is because both entities cannot maximise their payoffs by inclining to other strategies other than taking the (C, c) strategy. This equilibrium is achieved in an extended form of the game theoretic approach with perfect information.

Applying the auction-based pricing equations [3-4] to the above-mentioned application, results in the following figures for the base station and the cluster-head:

Table 1: Processing times

	Base Station	Cluster-head
Available Processing Time (P_a)	5000	1000
Pre-deployed Task’s Required Processing Time (P_d)	2500	500
New Task’s Processing Time (P_t)	2500	500
Query/Response Processing Time (P_q)	500	100
Total Energy (E_{total})	N/A	100,000

It is worth noting that the actual processing time in Sensomax’s architecture has been defined in millisecond unit. However, for simplicity the above figures are normalised by a factor of 1000000.

Once nodes’ utilities and tasks’ prices have been calculated using the aforementioned auction-based techniques, the rewards gained by the peers, based on the notations given in table 1, can be calculated using the following functions:

$$E_{saving} = P_a - \left[\left(\frac{\sum_{i=0}^{i=n} P_{d_i}}{n} + \frac{\sum_{j=0}^{j=m} P_{t_j}}{m} \right) + \sum_{k=0}^{k=s} P_{q_k} \right] \quad (1)$$

This function simply returns how much energy can be saved by taking into account the number of pre-deployed tasks (n), new tasks (m) and the total number of query/responses (k). Having calculated the total saved energy, node’s remaining energy can be calculated by deducting the saved energy from the total energy.

For the purpose of this section, we will not deal with the remaining energy, and the only focus will be on the saved energy, which is considered as the reward. Based on the actual pricing units, which were shown in table 1, the base station will compare its available strategies, whilst calculating the following rewards using function 1. The calculated rewards are therefore shown in table 2.

Table 2: Actual rewards for the base station and cluster-head's strategies

CH \ BS	A	B	C	D
a	-1000 / 0	0 / -700	1250 / -700	-500 / -700
b	-1000 / 0	0 / -500	1250 / -500	-500 / -500
c	-1000 / 0	0 / 750	1250 / 750	-500 / 750
d	-1000 / 0	0 / 0	1250 / 0	-500 / 0

Table 2 represents the actual rewards (profit and loss) of the base station and the cluster-head in normalised millisecond units. As in table 2, strategy ‘C’ for the base station and strategy ‘c’ for the cluster-head result in maximum rewards. According to this table, the pre-deployed task requires 2500 processing time, whereas the new task requires the same amount. Therefore as the result taking options ‘A’ by the base station and option ‘a’ by the cluster-head, both entities suffer profit loss of responding unnecessary queries from their higher-ranking peers. First query request the deployed task’s status, and the second one is the entities’ response to the query. That results in two send and receive queries for each node, costing 500 and 100 per query/response for the base station and the cluster-head respectively. In addition, the cluster-head also suffers a major 500 loss for not executing the received task. Taking option ‘B’ by the base station results in querying the cluster-head immediately, at the same time as the pre-deployed application. That results in no profit or loss for the base station as the gain of 500 profit for the second application evens out the loss of 500 for the pre-deployed one. For the cluster-head however, taking strategy ‘b’ results in ignoring the pre-deployed application’s requirement, thus suffering 500 loss. If the base station takes strategy ‘C’, it can save a total of 1250 as it serves both applications fairly by summing up half the utilities of the pre-deployed application with the new one [Saving = (2500/2) + 2500]. The base station’s strategy ‘C’ leaves the base station a profit of 1250 The Cluster-head’s Strategy ‘c’ results in serving both applications, whilst causing slight delay in processing the first one, thus achieving a gain of 750 [saving = (500/2) + 500]. Taking strategy ‘D’ will gain the base station with loss of 500 as the new application is served with latency, thus not meeting the end-user’s requirement. The cluster-head’s loss however, evens out the gain by the pre-deployed one, thus resulting in no loss or profit.

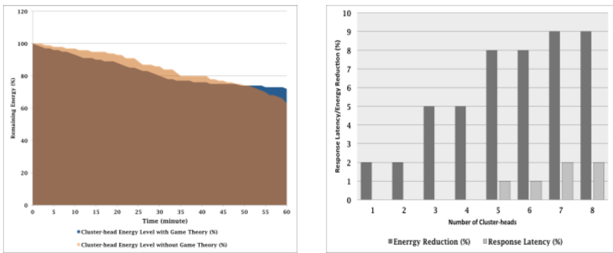
What we have described so far only included the interaction between the base station and a single cluster-head. In order to expand the game to involve more players, the base station iterates the same process for every cluster-head involved in the task distribution process.

IV. CASE STUDIES AND EVALUATION

In [18] we demonstrated the energy profiling and packet loss of nodes and cluster-heads using game theory. Here we will review those results, as well as investigating latency and cluster density impacts on different operational paradigms with and without game theory utilization. For the first part of each case study we will reiterate the same experiment, which was thoroughly explained in [18].

In our first experiment, we tried to validate how effectively game theory can contribute towards energy consumption between the base station and cluster-heads. This was mainly done to optimise task distribution as the first step in which cluster-heads receive their tasks. As we mentioned in the previous section, task allocation can be challenging, depending on cluster-heads' properties, such as their remaining energy and pre-deployed tasks. Therefore we have built two separate applications. The first one is deployed and executed as the pre-deployed application, and the second application is deployed whilst the first one is still running. This approach creates a situation, where nodes tend to compete, in order to take the new application's tasks depending on their available resources as the result of executing the first application's tasks.

The first application is a time-driven application, which demands light level sensory data every second upon base station's query (request), as well as requiring the sensor node to report temperature every 10 seconds without any request from the base station. The second application demands light level every 5 seconds upon base station's request, as well as requesting automatic reporting of acceleration on three-axis (X,Y,Z) every 500 milliseconds. Cluster-heads receiving these applications query their members for the required parameters at the specified intervals. This experiment is repeated twice, with and without the game theoretic approach involved in the execution process.

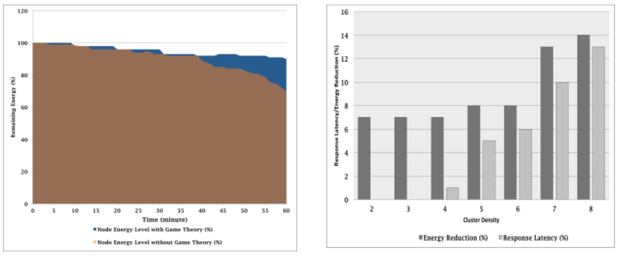


A B

Figure 2: (A) Cluster-head energy profiling with and without game theory [18] (B) Energy reduction and latency associated with the number of cluster-heads using the game theoretic approach

Based on the explanation given in the previous section and how cluster-heads decide on allocating different priorities to their given tasks, the results achieved from the experiment are shows in figure 2(A).

As a result of such trial and error iterations, and mainly due to the high number of price calculations, the energy level drops significantly in the beginning of the process. However, once a winning strategy is chosen, energy-hungry price calculating process is reduced considerably, and the cluster-head tends to stay with a single strategy. Therefore as brown histogram shows, cluster-head's energy spending stabilises after ~30 minutes. In fact, the cluster-head achieves a better energy profiling compared to the non-game theoretic approach (blue histogram). Figure 2(B) demonstrates the reduction in the energy consumption and the latency of 1-10 cluster-heads with and without the game theoretic approach in a wireless sensor network. The applications used in this experiment are the same applications used in the previous experiment, which were deployed in the same fashion. As figure 2(B) shows, the dark grey bars denote the total reduction in the energy consumption of the network compared to the non-theoretic approach, whereas the light grey bar represent the latency caused in the cluster-heads response to the base station during the lifetime of the applications. As dark grey bars show in this figure, the higher number of cluster-heads (players), the higher the energy reduction becomes. That is mainly due to the higher number of cluster-heads fulfilling the application, whereby game theory can facilitate task distribution amongst higher number of cluster-heads.



A B

Figure 3: (A) Sensor nodes' energy profiling [18], (B) Energy reduction and latency associated with different cluster densities using game theoretic approach

The presence of more players, helps the base station to locate cluster-heads faster. Mainly because they already are engaged in other tasks. Whereas in the non-game theoretic mechanism, tasks are simply split up amongst cluster-heads with no consideration on the overhead imposed on them whilst executing other tasks.

This outcome prompts the end-users to adopt more cluster-heads in order to reduce the energy consumption. However, as we mentioned earlier, as the number of players are increased, the timely response of the cluster-heads are reduced. As the first three light grey bars on the far left side of the chart show, the response delay for 1-3 cluster-heads is around 1%, which can be considered insignificant. That trivial impact is with regards to the second application, where three acceleration variables (X, Y, Z) are reported every 500ms, which could sum up to 15ms. However, as the number of cluster-heads increase, the latency could increase to up to 4%, which considered quite vital, over the lifetime of the application (i.e. 4% latency for reporting the acceleration over an hour period is equivalent to nearly 216 seconds).

In this experiment, each cluster-head was allocated two members only. The other factor we tried to focus on was how cluster density affects both energy consumption and response latency within each cluster . Therefore we repeated a style of the previous experiment, this time with a variable number of nodes in each cluster.

As figure 3(B) shows, the dark grey bars represent the total game theoretic energy reduction with a variable number of nodes in a single cluster compared to the non-theoretic approach. According to this figure, as with the higher number of cluster-heads, the higher number of nodes in a cluster reduces the energy consumption significantly. The total reduction energy consumption reaches up to 13% with 8 members in a cluster, which is nearly a third higher the quantity of energy saved over 8 cluster-heads. However, the response latency is also considerably higher. Based on the figure 2(B) and 3(B), the optimal number of cluster-heads and cluster densities used in a WSN needs to be in the range of 1-3 cluster-heads, each containing 2-3 members, in order to achieve a reasonable latencies towards meeting application requirements.

Figure 3(B) shows the average energy profiling of 9 nodes, where each 3 nodes report to a single cluster-head As the brown histogram shows, the game theoretic approach can save nearly 10% on the energy consumption with our experimental optimal values [18].

The experiments reported in this section were mainly conducted using time-driven applications. The next experiment will investigate how game theoretic approach affects energy consumptions of applications with different operational paradigms as in table 3.

Table 3: Applications with different operational paradigms

Application	Operational Paradigm	Frequency/ Threshold	Parameter
A	Query-driven	250ms	Light and Temperature
B	Data-driven	>300ms	Light and temperature over 5 minutes
C	Event-driven	>500 <20	Light
D	Time-driven	5 seconds	Light and Temperature

Previous experiment was repeated four times, each time deploying one of the above-mentioned applications. Figure 4 shows the energy profiling of different applications with different operational paradigms, with (black bars) and without (green bars) the game theoretic approach.

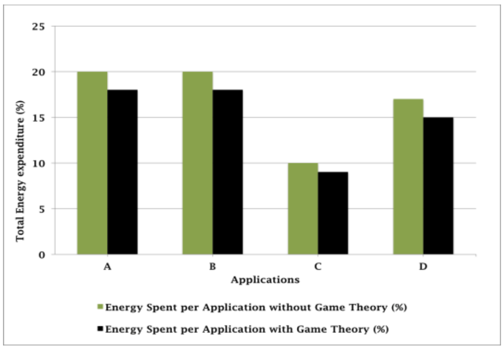


Figure 4: Energy expenditure for different operational paradigms

According to figure 4, except application C, which is an event-driven application, game theoretic approach saves around 2% on the total energy consumption of all operational paradigms. That is because, event-driven applications involve various unexpected events, which are triggered according to the given environment. Therefore it leaves less flexibility to the cluster-heads' game theoretic mechanism in order to stabilise and adapt to the application's behaviour.

V. CONCLUSION

In this paper we have shown how utilizing game theory can improve energy consumption for distributing tasks amongst sensor nodes. Sensomax, as a multitasking WSN middleware, exploited the game theoretic approach in conjunction with auction-based techniques, in order to identify the available strategies involved in tasks distribution process. The proposed game theoretic approach helped Sensomax to allocate resources to the deployed applications, based on nodes' processing and memory availability, as well as their remaining energy level. We demonstrated how energy consumption may be reduced in a multitier, hierarchical WSN, where applications are collaboratively executed by multiple clusters.

ACKNOWLEDGMENT

This work was performed under the SPHERE IRC, funded by the UK Engineering and Physical Sciences Research Council (EPSRC), Grant EP/K031910/1.

REFERENCES

[1] M. Haghighi, D. Cliff, "Sensomax: An Agent-Based Middleware For Decentralized Dynamic Data-Gathering in Wireless Sensor Networks", The 2013 International Conference on Collaboration Technologies and Systems, CTS 2013, May 2013, San Diego, USA.

[2] M. Haghighi, D. Cliff, "Multi-Agent Support for Multiple Concurrent Applications and Dynamic Data-Gathering in Wireless Sensor Networks", The Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS-2013, July 2013, Taiwan.

[3] M. Haghighi, "Market-based Resource Allocation For Energy-efficient Execution of Multiple Concurrent Applications In Wireless Sensor Networks". International Conference on Ubiquitous context-awareness and Wireless Sensor Networks, South Korea, July 2013.

[4] M. Haghighi, "Cooperative Task Allocation in Utility-based Clustered Wireless Sensor Networks", International Journal of Information and Electronics Engineering, (IJIEE), September 2013.

[5] H.-Y. Shi, W.-L. Wang, N.-M. Kwok, and S.-Y. Chen, "Game Theory for Wireless Sensor Networks: A Survey," *Sensors*, vol. 12, pp. 9055-9097, 2012.

[6] T. Spyridopoulos, G. Oikonomou, T. Tryfonas, and M. Ge, "Game Theoretic Approach for Cost-Benefit Analysis of Malware Proliferation Prevention," in *Security and Privacy Protection in Information Processing Systems*. vol. 405, L. Janczewski, H. Wolfe, and S. Shenoi, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 28-41.

[7] M. Asadi, C. Zimmerman, and A. Agah, "A Game-theoretic Approach to Security and Power Conservation in Wireless Sensor Networks," *International Journal of Network Security*, vol. 15, pp. 50-58, 2013.

[8] S. Sengupta, M. Chatterjee, and K. A. Kwiat, "A Game Theoretic Framework for Power Control in Wireless Sensor Networks," *Computers, IEEE Transactions on*, vol. 59, pp. 231-242, 2010.

[9] E. Campos-Nañez, A. Garcia, and C. Li, "A Game-Theoretic Approach to Efficient Power Management in Sensor Networks," *Operations Research*, vol. 56, pp. 552-561, 2008.

[10] X. Dongfeng, S. Qi, Z. Qianwei, Q. Yunzhou, and Y. Xiaobing, "An Efficient Clustering Protocol for Wireless Sensor Networks Based on Localized Game Theoretical Approach," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[11] G. Koltsidas and F.-N. Pavlidou, "A game theoretical approach to clustering of ad-hoc and sensor networks," *Telecommunication Systems*, vol. 47, pp. 81-93, 2011/06/01 2011.

[12] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, 2000, p. 10 pp. vol.2

[13] M. Ayers and L. Yao, "Gureen Game: An energy-efficient QoS control scheme for wireless sensor networks," in *Green Computing Conference and Workshops (IGCC), 2011 International*, 2011, pp. 1-8.

[14] M. L. Tsetlin, Linear Automaton Theory and Modeling of Biological Systems. New York, 1973.

[15] R. Hongliang and M. Q. H. Meng, "Game-Theoretic Modeling of Joint Topology Control and Power Scheduling for Wireless Heterogeneous Sensor Networks," *Automation Science and Engineering, IEEE Transactions on*, vol. 6, pp. 610-625, 2009.

[16] Y. B. Reddy, "A Game Theory Approach to Detect Malicious Nodes in Wireless Sensor Networks," in *Sensor Technologies and Applications, 2009. SENSORCOMM '09. Third International Conference on*, 2009, pp. 462-468.

[17] Maraslis, K., Spyridopoulos, T., Oikonomou, G., Tryfonas, T., & Haghighi, M. (2015). Application of a Game Theoretic Approach in Smart Sensor Data Trustworthiness Problems. In *ICT Systems Security and Privacy Protection* (pp. 601-615). Springer International Publishing.

[18] M. Haghighi, K. Maraslis, T. Tryfonas, G. Oikonomou, "Game Theoretic Approach Towards Energy-Efficient Task Distribution in Wireless Sensor Networks", IEEE SENSORS 2015, Busan, South Korea, November 2015.